Final Exam — ECE 438

Due May 16^{th}, 2020

Name:

- This is a open-book, open-note exam.
- You *must* work on this exam by yourself!
- Calculators are permitted, but no communication devices of any kind are allowed.
- Each question is marked with its number of points.
- Show your answers in the space provided for them. Write neatly and be well organized.
- Show your work if you want to get credit!

Good luck!

Problem	Maximum	Score
1	15	
2	15	
3	20	
4	20	
5	20	
6	15	
7	15	
Total	120	

1. **(15 points)** You are looking to accelerate public-key encryption and found through simulation that 95% of the execution time of a particular algorithm is spent performing modular multiplication. Assume for the following problems that the execution time of your algorithm is 2 seconds without hardware acceleration.

   (a) What would be the minimum achievable execution time of this same algorithm if you developed hardware that could accelerate just the modular multiplication by 50x? Calculate the overall speedup.

   (b) What would be the execution time and speedup achievable with a 20x acceleration of modular multiply?

   (c) What is the name given to the design principle used in parts a) and b)?

   (d) What would the theoretical maximum speedup be for your program if you could infinitely accelerate modular multiply?

2. **(15 points)** Imagine you have a 1GHz RISC processor with split L1 instruction and data caches, each 16kB in size with a hit time of 1ns. Access to main memory takes 50 ns, and 35% of instructions access memory. The L1 instruction cache miss rate is 1%, while the L1 data cache miss rate is 4%.

   (a) Calculate the Average Memory Access Time (AMAT) for each of the L1 caches.

   (b) Assuming your processor has a CPI of 1.3 with an ideal memory hierarchy, what is the CPI considering memory stalls?

   (c) You are considering the inclusion of a 256kB L2 cache to improve your performance. If the miss rate of the L2 is 35%, what would the AMAT of the L1 instruction and data caches be with the L2 cache? Assume a 6ns L2 hit time.

   (d) What would the CPI of your processor be with the L2 cache? What is the speedup due to the L2 cache?

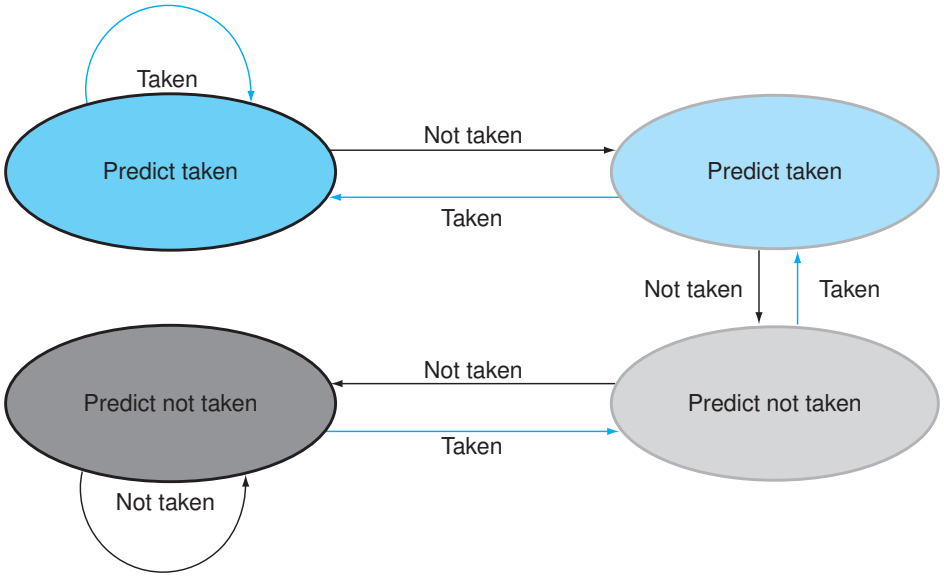
3. **(20 points)** For the following problem, assume a 5-stage pipelined processor with a branch delay slot and branch resolution in the Execute stage. Consider the code below:

```
    addi $t2, $t1, 48
loop:
    lw $t4, 40 ($t1)
    lw $t5, 80 ($t1)
    add $t6, $t4, $t5
    sw $t6, 120 ($t1)
    bne $t1, $t2, loop
    addi $t1, $t1, 4
    add $t2, $t1, $t3
```

(a) Draw the pipeline execution diagram for the first *two* iterations of the above code when an "assume not taken" branching scheme is used.

(b) How many clock cycles are required to execute the above code to completion when an "assume not taken" branching scheme is used?

(c) How many clock cycles are required to execute the above code assuming a 100% correct branch predictor.



(d) Consider the use of the two-bit predictor shown above. Assuming the predictor starts in the bottom right state (*different than homework*), how many clock cycles are required to execute the above code?

(e) What is the accuracy of this predictor given the code above? Compare that to the accuracy of the "assume not taken" scheme.

(f) What speedup does the branch predictor from part (d) provide over the "assume not taken" scheme?

4. **(20 points)** The following questions are related to pipelining and instruction level parallelism:

   (a) Compare the i486 pipeline with that of the MIPS pipeline we studied in lecture. How does it differ? How is it similar? How did the performance compare?

   (b) How did the Intel Pentium achieve a performance benefit over the i486? How did the performance compare?

   (c) Describe a tournament predictor. How does it provide a benefit over a local or global predictor?

   (d) Define precise exceptions. How do we maintain precise exceptions in an inorder pipeline? How do we do so in an out-of-order machine?

5. **(20 points)** The following questions are related to cache design and memory hierarchy:

   (a) Briefly describe the types of cache misses talked about in lecture. Remember the three C's.

   (b) How does adding multiple levels to the memory hierarchy improve performance?

   (c) Why might one add *associativity* to a cache? How might one add associativity? What are the drawbacks to set-associative caches?

   (d) Why might the instruction cache in a *split cache* system have a lower miss rate than the data cache?

   (e) Describe the purpose of a *write buffer*. Where in the memory hierarchy might you place a write buffer?

   (f) Briefly describe *write-back* and *write-through* cache write policies. When might one be preferred over the other?

(g) Describe each of the following architectures: Von Neumann, Harvard, Modified Harvard.

6. **(15 points)** The following questions are related to virtual memory:

(a) What was the primary reason virtual memory was developed?

(b) What are other advantages virtual memory provides?

(c) What is the purpose of the TLB? How do the TLB and L1 cache work together?

(d) What role does the Operating System (OS) play when it comes to virtual memory? What additional hardware support is required to run an OS?

7. **(15 points)** Assuming a direct-mapped cache with a byte-address that is broken up such that bits 0-2 are for the byte offset, bits 3-5 are for the word offset, bits 6-14 are for the index, and bits 15-31 are for the tag, answer the following questions:

   (a) How large are the words in this machine?

   (b) How many words are in each cache block? How many bytes are in each cache block?

   (c) How many cache blocks are in the cache? How many sets are in the cache?

   (d) How large is the data store of this cache?

   (e) How large is the tag store? Assume a *valid* and *dirty* bit are included with each tag.

   (f) If you were to modify the cache to be 4-way set associative (*different from homework*) but keep the data store the same size, what size would the tag and index be? How large would the tag store be?

   (g) How much memory can the above machine address?